

ARexx-Makros  
für  
PasT<sub>E</sub>X und GoldEd

Axel Burghardt

22. März 1994

## 1 AR<sub>EXX</sub>-Makros für PasT<sub>EX</sub> und GoldEd

Das AR<sub>EXX</sub>-Makro `AutoDoc.rexx` erlaubt jedem Programmierer AutoDoc-Dateien zu erstellen. Der Konverter AD2AG, der aus AutoDoc-Dateien AmigaGuide-Dateien erstellt, wurde bereits auf der AMIGA Plus Diskette 7/93 veröffentlicht.

Zwei AR<sub>EXX</sub>-Makros erleichtern speziell das Arbeiten mit GoldEd.

Die übrigen Makros erlauben das Steuern von PasT<sub>EX</sub> durch GoldEd.

Die Makros wurden erstellt und getestet auf

- AMIGA 4000/40, Kickstart 39.106, Workbench 39.29
- AR<sub>EXX</sub> V1.15
- GoldEd/Pro V0.98
- PasT<sub>EX</sub> V3.14
- ShowDVI V1.16L

Mein besonderer Dank richtet sich an

- Dietmar Eilert (GoldEd)
- Georg Heßmann (Pas-T<sub>EX</sub> und AR<sub>EXX</sub>-Skripte)
- Jörg Höhle (AR<sub>EXX</sub>-Skripte)

**AutoDoc.rexx:** bietet jedem die Möglichkeit, AutoDoc-Dateien zu erstellen. Im Modus AUTODOC ist das Skript kompatibel mit AutoDoc V1.0 aus dem NDUK<sup>1</sup> mit dessen Optionen `-C -I`. Es extrahiert dann aus dem Quelltext (C-Source) die entsprechenden AutoDoc-Einträge.

Wird AUTODOC als Parameter *nicht* übergeben, läßt das Skript die AutoDoc-Einträge in ihrem Kontext stehen. Der große Vorteil besteht darin, daß der **komplette** Quelltext innerhalb der Hypertext-Struktur vorliegt und somit auch Verweise (Links) zu Funktionen/Strukturen (Nodes) möglich sind, sofern diese in den AutoDoc-Köpfen<sup>2</sup> eingesetzt werden.

Zur Installation kann das Makro nach **REXX:** kopiert werden. Eine andere Möglichkeit besteht darin, das Makro als Menüpunkt in GoldEd (mit oder ohne Parameter) einzubinden. Dazu sollte es jedoch in das Verzeichnis **GOLDED:AREXX** kopiert werden.

**LoadSpecialConfig.ged:** öffnet eine spezifische Konfigurationsdatei. Die Auswahl der Konfiguration erfolgt über die Endung der Datei, die GoldEd zum Öffnen übergeben wird. Der Kunstgriff in der **while**-Schleife ist notwendig, damit das 'LOCK' erst erfolgt, nachdem die Datei geladen ist.

Es können so für beliebige Datei-Klassen (.T<sub>EX</sub>, .C, .TXT usw.) benutzerspezifische Konfigurationen erstellt werden.

Zur Installation wird der Skript-Name unter **Konfig/Diverses** im Feld **Startup-Makro** eingetragen und die Einstellung unter **GoldEd.prefs** gesichert.

---

<sup>1</sup>Native Developer Update Kit

<sup>2</sup>Ein Standard-C-Kopf liegt auf Diskette bei.

Wurden alle Konfigurationen wie gewünscht eingestellt, darf das globale Sichern über `Konfig/Sichern...` nicht vergessen werden.

**DocLine.ged:** hilft beim Dokumentieren von Quelltext in einfachster Form. Es eröffnet einen Requester und erfragt den Kommentar. Die gewünschte Spaltenanzahl kann dem Makro als Parameter übergeben werden (z.B. `DocLine.ged 70`). Der Standardwert beträgt 78 Spalten (gemäß `AutoDoc`).

Das Ergebnis sieht dann z.B. so aus:

```
/* ----- Ein Kommentar zum Quelltext ----- */
```

Gedacht ist das Makro als zusätzlicher Menüpunkt für `GoldEd`.

**TEX-Server-Start.ged:** ist ein Makro des Makro-Pakets, mit dem `PasTEX` von `GoldEd` gesteuert werden kann.

Alle Makros sind in das Verzeichnis `GOLDED:AREXX` zu kopieren und können unter einem Menüpunkt (z.B. `PasTEX`) installiert werden.

Der `TEX`-Server spielt dabei die wichtigste Rolle. Er kontrolliert den `TEX`-Übersetzer `VirTEX`, ruft `ShowDVI` oder ggf. den Editor, sofern der Übersetzungsvorgang nicht fehlerlos war.

Bisher versteht der Server (`TEX.1`) die Kommandos `SHOWDVI`, `QUIT` und `COMPILE`.

**TEX-Server-Stop.ged:** beendet den laufenden `TEX`-Server.

**TEX-Start.ged:** koppelt den `TEX`-Quelltext von `GoldEd` ab und reicht ihn über das `COMPILE`-Kommando an den `TEX`-Server zur Übersetzung weiter.

Sollte der `TEX`-Server noch nicht aktiv sein, wird dieser auf Wunsch nachträglich gestartet. Die Übersetzung erfolgt jedoch erst nach erneutem Makro-Aufruf.

**TEX-Edit.ged:** ist für die Fehlerbehandlung verantwortlich. Ist bei der Übersetzung ein Fehler aufgetreten, wird das *Logfile* auf Wunsch in den Editor geladen. Ein bereits geöffnetes *Logfile* wird dabei überschrieben. Im *Logfile* wird auf den ersten Fehlertext, im `TEX`-Text auf die fehlerhafte Stelle positioniert.

**TEX-Error.ged:** kann von Fehler zu Fehler springen. Die Richtung (`PREV` oder `NEXT`) kann als Parameter übergeben werden. Der Standard-Wert ist `NEXT`. Wird ein Fehler in der angegebenen Richtung gefunden, erfolgt die Positionierung wie bei `TEX-Edit`.

**GED2Front.ged:** bringt den Screen von `GoldEd` nach vorne. Das Makro sollte in der Konfigurationsdatei<sup>3</sup> von `ShowDVI` auf eine Funktionstaste gelegt werden.

**TEX-DVI.ged:** ist das Gegenstück zu `GED2Front`. Es veranlaßt das Laden der `DVI`-Datei nach `ShowDVI`. Die Steuerung erfolgt durch den `TEX`-Server (Kommando: `SHOWDVI`). Dieser bringt dann automatisch den Screen von `ShowDVI` nach vorne.

**TEX-Formblatt.ged:** ist ein Makro, das dem `TEX`-Anwender viel Arbeit abnehmen kann. Es stellt Formblätter zur Verfügung (z.B. `Brief`) und hilft bei deren Ausfüllen.

Das bisher implementierte Formblatt `Brief` unterstützt das Ausfüllen von Adresse und Anrede. Ein Standard-Formblatt `Brief.TeX`, sowie mein persönlicher Brief-Style liegt als Anregung bei.

---

<sup>3</sup>Default: `TEX:config/showdvi.config`

Das Formblatt sollte in das Verzeichnis `TEXT:Formblatt` kopiert werden. Ansonsten ist der Pfad im Makro anzupassen.

Ich werde die Makros bei Bedarf erweitern oder verbessern. Fehler und Wünsche können somit schriftlich an mich<sup>4</sup> gerichtet werden.

Ein ähnliches Makro-Paket zur Anbindung von GNU C/C++ ist in Vorbereitung.

---

<sup>4</sup>Axel Burghardt, Hohlweg 11, 53773 Hennef